Interactive Common Illumination for Computer Augmented Reality

George Drettakis *, Luc Robert **, Sylvain Bougnoux **

* *i*MAGIS/GRAVIR-INRIA, ** ROBOTVIS

Abstract: The advent of computer augmented reality (CAR), in which computer generated objects mix with real video images, has resulted in many interesting new application domains. Providing *common illumination* between the real and synthetic objects can be very beneficial, since the additional visual cues (shadows, interreflections etc.) are critical to seamless real-synthetic world integration. Building on recent advances in computer graphics and computer vision, we present a new framework to resolving this problem. We address three specific aspects of the common illumination problem for CAR: (a) simplification of camera calibration and modeling of the real scene; (b) efficient update of illumination for moving CG objects and (c) efficient rendering of the merged world. A first working system is presented for a limited sub-problem: a static real scene and camera with moving CG objects. Novel advances in computer vision are used for camera calibration and user-friendly modeling of the real scene, a recent interactive radiosity update algorithm is adapted to provide fast illumination update and finally textured polygons are used for display. This approach allows interactive update rates on mid-range graphics workstations. Our new framework will hopefully lead to CAR systems with interactive common illumination without restrictions on the movement of real or synthetic objects, lights and cameras.

1 Introduction

Computer augmented reality (CAR) is a booming domain of computer graphics research. The combination of virtual or synthetic environments with real video images (RVI) has lead to many new and exciting applications. The core research in this area concentrates on the problems related to registration and calibration for real-time systems (see for example [2, 3]). Since many of these problems are still largely unresolved, little attention has been given to the problems of the interaction of *illumination* between the real and synthetic scenes.

Pioneering work in this domain has been performed by Fournier et al. [14]. This work (see Section 2.3 for a brief review), has shown how the computation of common illumination between the real and synthetic scene results in a greatly improved graphical environment with which the user can interact. The use of real video images eliminates the need to model complex environments in great detail, and, by nature, provides a realistic image to the user. In what concerns common illumination, the introduction of virtual objects in a real scene becomes much more natural and convincing when light exchanges between real and synthetic objects (such as shadows and interreflections) are present in the composite images presented to the user.

In this work we present a new common illumination framework, by addressing the following three stages: (a) camera calibration and modeling, (b) common illumination

* iMAGIS is a joint research project of CNRS/INRIA/INPG/UJF. Postal address: B.P. 53, F-38041 Grenoble Cedex 9, France Contact E-mail: George.Drettakis@imag.fr

** INRIA, BP93 06902 Sophia-Antipolis, Cedex, France, E-mail: Luc.Robert@inria.fr

updates and (c) rendering. The goal is to build a system which can compute common illumination at interactive update rates. The work reported here is in preliminary form; as such we have restricted the configuration we will be treating to the case of moving computer generated objects in a static real scene viewed by a static camera.

By using advanced vision techniques, we have replaced the tedious and inaccurate manual modeling process with a flexible and precise vision-based approach. This method allows us to model the real scene to the level of detail required, and to extract camera parameters simply and automatically. We use fast hierarchical [16, 24, 25] and incremental update [8] techniques for radiosity, permitting interaction with virtual objects in the CAR environment. Interactive update rates (a few seconds per frame) of the mixed real/synthetic environment, including common illumination is achieved by using a texture-based rendering approach on suitable hardware. We believe that the combination of advances in vision, illumination and graphics provides a framework which will lead to general interactive common illumination for CAR.

2 Previous and Related Work

2.1 Reconstruction of 3D models From Images

A number of techniques have been proposed for producing 3D models from images in photogrammetry and computer vision. The photogrammetry approach mostly focuses on accuracy problems, and the derived techniques produce three-dimensional models of high quality [1]. However, they generally require significant human interaction. Some commercial products, such as *Photomodeler*, already integrate these techniques. In computer vision, a number of automatic techniques exist for computing structure from stereo or motion (e.g., [7, 20, 10]). With these techniques, the three-dimensional models are produced much more easily, but they are less accurate, potentially containing a small fraction of gross errors.

Alternate representations have been proposed for realistic rendering from images. With image interpolation techniques [11, 21, 23], the scene is represented as a depth field, or equivalently, as a set of feature correspondences across two reference images. Although these implicit 3D representations are suited to rendering, they are not adapted to our framework since we need *complete* 3D data to perform radiosity computation.

Some recent approaches have been proposed to reduce the effort in the production of explicit 3D models of high quality, either by imposing constraints on the modeled scene [6], or by combining automatic computer vision processes with human interaction [12]. We follow this last approach in this paper.

2.2 Computer Augmented Reality

Much work has recently been performed in the domain of computer augmented reality. The main body of this research concentrates on the requirements of real-time systems [2]. In terms of illumination, these systems provide little, if any, common lighting information. Examples of work including some form of shadowing between real and synthetic objects are presented in [26] and [19].

Common illumination requires full 3D information, and thus should use explicit modeling of the real world objects. Similar requirements exist for the resolution of occlusion between real and virtual objects (e.g., [3]).

The wealth of excellent research in this domain will undoubtedly be central in the future work in common illumination (see Section 6.1). For now however, we concentrate

on the issues directly related to illumination. The reader interested in an in-depth survey should refer to [2].

2.3 Radiosity and Common Illumination for CAR

In what follows, we consider the following configuration: we have an image I, which we call the "target image", and, using techniques developed below, a set of geometric elements approximating the scene. All quantities related to the image will be noted " $\hat{.}$ ". The most closely related previous research in common illumination is that of Fournier et al. [14]. We will be adopting many of the conventions and approximations used in that approach. In [14] many basic quantities are defined in a rather ad-hoc manner using information taken from image I. The average reflectivity of the scene $\hat{\rho}$ is selected arbitrarily. This can also be set as the average pixel value.

Once a value for $\hat{\rho}$ is set, the *overall reflectivity factor* R is defined as:

$$R \;=\; \frac{1}{1-\hat{\rho}}\;. \tag{1}$$

The concept of "ambient radiosity" [5], $\hat{B}_A$ is then used, permitting a first estimation of the exitance values E_i of the sources:

$$\hat{B}_A = \frac{R\sum_{all\ i} E_i A_i}{\sum_{all\ i} A_i}, \tag{2}$$

where E_i is the exitance of each object i and A_i its area. Another approximation of $\hat{B}_A$ is given by:

$$\hat{B}_A = \frac{\sum_{all\ xy} p_{xy}}{N\hat{\rho}}, \tag{3}$$

where p_{xy} is the intensity of the pixel xy of the target image I, and N the total number of pixels of I. Equations (2) and (3) allow us to approximate the values of E_i if we know the number and area of the real sources.

Fournier et al. also proposed a first approximation of the radiosity on each geometric element i, which we call $\hat{B}_i$, which is the average value of the pixel intensities covered by element i.

In our approach, we improve the ease of modeling, as well as the lighting update and final display speeds compared to [14]. Nonetheless, to achieve these improvements, we sacrifice certain advantages of Fournier et al.'s system: we currently can only handle a static camera and real scene, and the quality of rendering may be slightly degraded compared to that obtained by ray-traced correction to a real image. Such degradation is mainly due to slight texture/polygon misalignment. However, since this paper is an attempt at defining a new approach to common illumination, we consider the above mentioned shortcomings as challenges for future research (Section 6.1).

3 Semi-Automatic Image-Driven Modelling Using Computer Vision

In this section we describe the creation of the three-dimensional model using vision-assisted techniques. We first compute the intrinsic parameters of the camera (focal length, aspect ratio) by using an image of a calibration pattern. Twelve images are then used to

automatically build a set of panoramic images. The relative positions/orientations of the cameras are then computed, based on point correspondences. We thus construct a geometric model of the room by computer-vision assisted, image-based interaction. Finally, textures are extracted and de-warped automatically. The whole process took approximately 4 hours for the scene shown in Figure 1.

3.1 Camera Calibration Using a Target

The intrinsic parameters of the camera (see [9] for more details about the imaging geometry of cameras) are computed using the calibration technique described in [22]. We need to take one image of a non-planar calibration pattern, i.e., a real object with visible features of known geometry. With minimal interaction (the user only needs to click the approximate position in the image of 6 reference points), an estimate of the camera parameters is computed. This estimate is then refined by maximising, over the camera parameters, the sum of the magnitudes of the image gradient at the projections of a number of model points.

The output of the process is a 3×4 matrix, which is decomposed as the product of a matrix of intrinsic parameters and a 4×4 displacement (rotation, translation) matrix (computation described in [9]).

3.2 Image Acquisition and Mosaicing

Though the minimum number of viewpoints for stereo reconstruction is two, we acquired images from four distinct viewpoints for better accuracy of the reconstructed 3D geometry. The viewpoints lie approximately at the vertices of a 1-meter-wide vertical square in one corner of the room.

To enlarge the field-of-view, we built panoramic images using mosaicing [27, 18]. At each viewpoint, we took three left-to-right images with an overlap of approximately 50% between two consecutive images. During this process, we were very careful at each viewpoint not to translate the camera but restrict motion to rotation. This guarantees that there exist linear projective transformations which warp the left and right images onto the center one.

For each triple of images, we computed these transformations automatically [29]. The two warped images and the center images were then "pasted" on the same plane. An example of mosaic is shown in Figure 1.

3.3 Computation of the Relative Geometry of the Cameras

In the next stage, we estimate the relative geometry of all the cameras, i.e., the rotations $\mathbf{R}_{1i}$ and translations $\mathbf{t}_{1i}$ of all cameras with respect to, say, the first one. For this, we identify corresponding points across the images. This is done in a semi-manual manner. Using the system `totalcalib` developed at ROBOTVIS (Figure 2), the user first clicks on a reference point in one image. The system then searches for matches in the other images, using window-based cross-correlation. This is shown in Figure 2 (a), with the annotated white points. The matches proposed correspond to the regions which are most similar to the image around the reference point. In most cases these points indeed represent the same object as the reference point. If not, the user can manually correct the errors.

Based on the point correspondences, we compute the fundamental matrices F_{1i} (see appendix 7) using the non-linear method described in [28]. The minimum number of

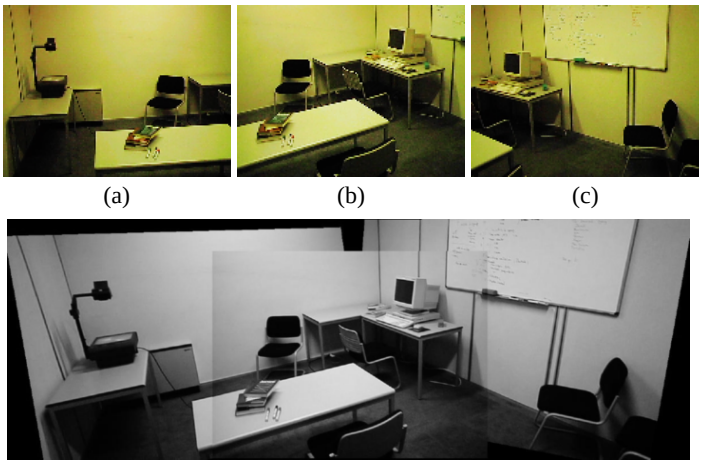
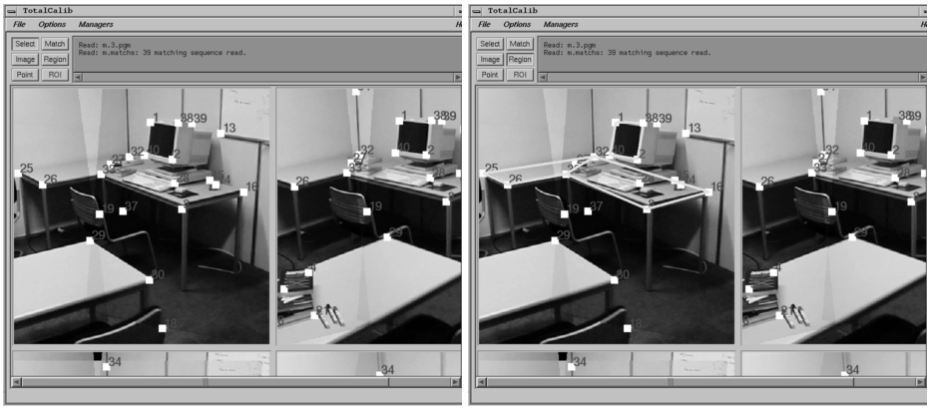


Fig. 1. Three original images, and the resulting mosaic (see text and also Colour Section).



(a) (b)

Fig. 2. (a) A `totalcalib` session; matched points are shown in white and are annotated. (b) Selection of the regions to reconstruct (e.g., the white polygon on the table-top).

correspondences is 8 in theory, but for better accuracy we used about 30 points spread over the whole scene (see Figure 2).

From F_{1i} and the intrinsic parameters, we then derive, using the technique described in [17], the rotation $\mathbf{R}_{1i}$ and translation $\mathbf{t}_{1i}$. In fact, each translation is known only up to a scale factor, which corresponds to choosing an arbitrary unit for distances in space. Translation $\mathbf{t}_{1i}(i > 2)$ is rescaled with respect to $\mathbf{t}_{12}$ by using point correspondences visible in images 1,2 and i and comparing space distances computed with image pairs $(1, 2)$ and $(1, i)$.

From this initial estimate, we then run a non-linear minimisation process known as *bundle adjustment* in photogrammetry [1], which refines the estimate of the rotations and translations. We end up with an estimate of rotations and translations of all cameras

with respect to the first camera.

3.4 Building the 3D Model and Extracting/De-warping the Textures

To build the polygons of the three-dimensional model, we first define the geometry of their vertices using the same semi-automatic technique. Their 3D coordinates are then obtained by inverting the projection equations. This process is known as *reconstruction* in computer vision or *intersection* in photogrammetry. We then manually define the topology of the polygons by selecting and connecting vertices in the images (see Figure 2(b)). The resulting model is stored in a standard 3D format.

For each polygon, we finally compute a texture image by de-warping the original image and bringing it back to the plane of the polygon. In this process, the resolution of the texture image can be chosen arbitrarily, as well as the directions of the axes of texture coordinates. The x-axis is chosen parallel to the longest edge of the polygon, which in most cases maximises the fraction of the texture image which lies inside the polygon and will be actually rendered. The choice of the texture resolution is based on the following criterion: when projecting one pixel of the texture image onto the reference image, one should obtain a small quadrilateral whose dimensions are all smaller than one pixel. This guarantees that the final synthesized images have approximately the same level of detail as the initial ones.

4 A Fast Hierarchical Method for Common Illumination

Recent advances in global illumination technology allow us to calculate the lighting efficiently, using hierarchical radiosity [16], clustering [24, 25] and incremental update methods [8]. To initialise the system, the calculation of certain basic parameters is required. We adopt many of the conventions used by Fournier et al. [14], adapting them appropriately to the application and the requirements at hand.

Two main stages are required: (a) initialisation of basic parameters such as exitance values for the real sources, radiosity and reflectance for the real video image (RVI) objects, and (b) the creation of a full hierarchical radiosity system, including the cluster hierarchy and "line-space" hierarchy of links and shafts required for the incremental solution.

4.1 Initialising the Basic Parameters

As discussed in Section 2.3 the basic approximations proposed in [14] can be used to estimate the set of parameters required to create a hierarchical representation of the (real) light transfer in the CAR scene. In the same spirit as this approach, we define the reflectance of each patch i to be: [3]

$$\hat{\rho}_i = \frac{\hat{B}_i}{\hat{B}_A} \times \hat{\rho} \tag{4}$$

Note that during subdivision, $\hat{B}_i$ is updated to reflect the average intensity of the pixels covered by the newly subdivided sub-element. The calculation of $\hat{B}_i$ is performed by rendering the polygon textured with the corresponding part of the target (real) image

[3] This is easier to calculate than the neighbourhood in [14].

I into an offscreen buffer and averaging the resulting pixel values. Once the new $\hat{B}_i$ is computed for the child element, the value $\hat{\rho}_i$ is updated.

It is important to note that this approach is a coarse approximation, since we cannot distinguish between shadows and obstacles in the image. Since we are simply computing an overall correction to illumination, we accept this approximation for now, but resolving this issue is definitely part of required future work.

Since we have an initial geometric model of the real sources, we can easily estimate their exitance. If (as is the case in the examples presented in Section 5), we have sources of equal power and area, the relations of equations (2) and (3) suffice to approximate E_i. If on the other hand we have a larger number of different sources, we need to estimate their value. This can be done easily by creating a link hierarchy using the $\hat{B}_i$'s and simply pulling $\hat{B}$ up the hierarchy. If we have m sources, by selecting m elements we have m equations giving us a good approximation of the E_i's.

4.2 Creating a Hierarchical Radiosity System

Once the values of E_i and ρ_i are estimated (we set $\rho_i = \hat{\rho}_i$ for each surface element), we have everything we need to perform a normal hierarchical radiosity iteration. Consider for example Figure 3(a), which shows the radiosity calculation for the real scene previously presented in Figure 1(b). Note that we only use *one* image of the mosaic (Figure 1(b) in our case) from which to extract textures.

The refinement stage of hierarchical radiosity proceeds as usual, and is left to run to "convergence", i.e. when the radiosity values no longer change much. Once completed, we have what we call an *original* value for the radiosities of all real objects. We store this value, $\tilde{B}_i$, on each hierarchical element (cluster, surface or sub-patch). The value $\tilde{B}_i$ is a (relative) representation of the illumination due to real sources.

The next step is the addition of computer generated objects. This is performed by adapting the methods described in [8]. We thus group the synthetic objects into "natural" clusters (i.e. a chair or a desk lamp) and we add them into the scene. To update the existing hierarchical radiosity system, we use the line-space traversal approach to efficiently identify the links affected by the CG object being inserted, and we incrementally perform the appropriate modification to illumination. As a result all patches now have a (possibly modified) radiosity value B_i.

4.3 Display

To achieve interactive update rates, we need to display the result of the combination of real and synthetic environments at interactive rates. This requirement precludes the use of the ray-casting approach of [14]. Our solution is to exploit the real-time texture capacities currently available on mid- and high-range graphics workstations.

To display the effects of a change due to the interference of a CG object with an RVI object, we simply modulate the texture by the ratio: $B_i/\tilde{B}_i$. This operation requires the capacity to modulate "positively" and "negatively", so we must coherently re-scale this ratio to always lie between zero and one.

4.4 Interactive Common Illumination of CG Objects in a Real Scene

Using an implicit hierarchical description of the line segment space contained between hierarchical elements, we can rapidly identify the links modified [8]. This is achieved by keeping a hierarchy of shaft structures [15] associated with the links and inactive (or *passive*) refined links.

In the case of the CAR application, special treatment is required to ensure that lighting effects created by CG objects are well represented. The refinement process is thus adapted to reflect this, by imposing finer subdivision for shadows or additional illumination due to the interaction of CG objects with the real scene (see Figure 3(c)).

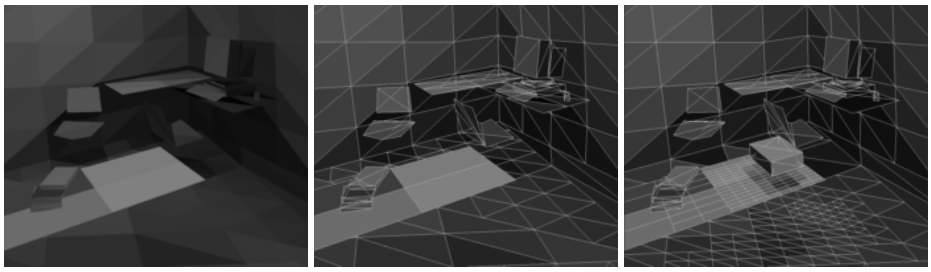


Fig. 3. (a) The radiosity $\tilde{B}$ computed by the initialisation phase of the algorithm. (b) The corresponding mesh. (c) The radiosity B and the mesh after the addition of the CG object.

Fig. 4. (a) The complete CAR rendering using RVI texture polygons for display, including the CG object. (b) The CG object moves to the left: update takes 2.5 seconds. (See Colour Section).

5 Results

The RVI scene we have used was modeled with 98 input polygons. This is a coarse representation, but sufficient for the example we wish to show here. We have a total of 4 512x512 textures (for the walls and floor), and 2, 2 and 6 textures of resolution 256x256, 128x128 and 64x64 respectively, for the detail objects of the scene.

In Figure 3(a) we show the result of the initialisation step where the geometry is displayed using the original radiosity $\tilde{B}_i$. Notice the low level of subdivision. The corresponding mesh is shown in Figure 3(b). After subdivision, the number of leaf elements is 512.

In Figure 4(a) we show the complete CAR image, including the CG object, and the corresponding shadow on the table top in the foreground. Notice how the mesh (Figure 4(b)) is much finer in the regions affected by the computer graphics object, with a total of 905 leaf elements.

The addition of the CG object took 2.8 seconds. When moving the dynamic object (see Figure 4), the update to illumination requires on average 2.5 seconds, on an Indigo 2, R4400 200Mhz High-Impact.

6 Future Work and Conclusions

The methodology we presented here was intended, as mentioned above, as a first step in a new direction for the treatment of common illumination for CAR. We thus consider it important to indicate why we believe that our framework is a suitable starting point for the treatment of more general configurations.

The ultimate goal is to have seamless, real-time mixing of real and synthetic scenes, with shared realistic illumination. There is a lot of work to be done before this goal can be achieved, much of which is related to hardware, vision, registration and sensing (see [2] for more detail). We concentrate here on the issues directly or indirectly related to common illumination and display.

6.1 Future Work

The first restriction to lift is that of a static camera. As a first step, we will be using pre-recorded real video sequences and attempt to mix real and synthetic scenes. Several problems result from this, notably camera calibration and correct rendering.

To deal with the problem of camera calibration, a first approach could be to use a set of "keyframes" for which the process described in this paper is applied, and to use point-tracking techniques to update the projective matrices as we move from one point to the other.

For rendering, the problem is posed by the fact that different de-warped textures will be associated with the same geometry at different viewpoints. Simple interpolation schemes will not work, since the occlusion configuration will have changed. Thus an obstacle elimination scheme will have to be developed, using known vision techniques enhanced with the available 3D and lighting information.

A different restriction to overcome is to permit motion of CG light sources. This requires an adaptation of the incremental update method [8], most notably in what concerns the representation of direct lighting shadows and corresponding refinement. The motion of real sources will result in similar problems.

Moving real objects is also an important challenge. In the context of pre-recorded sequences, much of the difficulty will be overcome by the explicit 3D modeling of the object. The removal of real objects is also an interesting challenge, and will require the use of some of the techniques developed for the treatment of the texture de-warping problem, in particular to effect a "removal" of a real object. Image-based rendering approaches [4] may prove useful here as well.

The move to real-time video acquisition and common illumination will be the greatest challenge of all. We believe that the knowledge and experience acquired in resolving the problem for pre-recorded sequences will prove extremely fruitful for the development of a solution working in real time.

6.2 Shortcomings of the Current Approach

Despite the encouraging first results, there are a several shortcomings in the approach presented here.

In the system presented we have not shown the addition of virtual lights. This is not too hard to achieve, but requires some modification to the incremental update approach, since the addition of a light source typically affects a large part of the environment. In addition, special attention must be taken in the re-scaling of the image before display since the addition of a source can add an order (or orders) of magnitude to the radiosity values of the scene.

In the lighting simulation phase we should use the variation of the RVI texture to aid refinement, and distinguish between variation due to occluding objects and shadows. The use of the obstacle removal techniques for textures can aid in this (see Section 6.1).

The refinement process is central: the modulation of the RVI textures by the changes in visibility is unforgiving. If a partially visible link is too far up the hierarchy, the resulting "spread shadow" is very visible, and spoils the effect of seamless real/synthetic merging. Since in synthetic-only environments these effects can usually be ignored, little has been previously done to address these issues.

The estimation of the initial parameters is also a major problem. The current estimations are very much ad-hoc. Nonetheless, what is important is not the precision of the approximation (since we are adding fictional objects, there is no "correct" solution), but the effect of more accurate choices which could result in more convincing results.

6.3 Conclusions

We have introduced a new framework for dealing with the problem of common illumination between real and synthetic objects and light sources in the context of computer augmented reality. We first use state-of-the-art vision techniques to calibrate cameras and estimate projection matrices, as well as recent image-based modeling approaches to create a model of the real environment. We then use rapid incremental hierarchical radiosity techniques to insert computer generated objects and manipulate them interactively. To achieve interactive display we use radiosity-modulated textures.

We have developed a working system for the restricted case of a static camera and static real environment. The prototype system we present shows that it is possible to create convincing CAR environments in which CG objects can be manipulated interactively. Compared to previous work in common illumination (notably [14]), our framework allows easier modeling and calibration, faster illumination updates and rapid display of CAR scenes.

Nonetheless, much more remains to be done. We have briefly discussed some possible future research paths, by removing the restrictions one by one, to achieve interactive common illumination for first a moving camera, then moving lights and finally moving real objects. We will initially be investigating these issues for pre-recorded video sequences, before taking the plunge into real-time acquisition.

In conclusion, we believe that the use of advanced, user-friendly image-based vision approaches to modeling and camera calibration, in conjunction with rapid incremental lighting and texture-based rendering, are a promising avenue leading to interactive common illumination for CAR. It will probably be a long time before we can interact naturally with virtual objects or creatures in our living room, but any solution to such a goal necessarily requires real-time common illumination.

Acknowledgements Thanks to Céline Loscos and François Sillion for carefully rereading the paper and for many fruitful discussions. Thanks to Frédo Durand for help with the video as well Rachel Orti, Mathieu Desbrun and Agata Opalach for final production.

References

1. K.B. Atkinson, editor. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, 1996.
2. Ronald T. Azuma. A survey of augmented reality. In *Presence: Teleoperators and Virtual Environments (to appear) (earlier version in Course Notes #9: Developing Advanced Virtual Reality Applications, ACM SIGGRAPH (LA, 1995), 20-1 to 20-38*, 1997. http://www.cs.unc.edu/ãzuma/azuma_AR.html.
3. D. E. Breen, R. T. Whitaker, E. Rose, and M. Tuceryan. Interactive occlusion and automatic object placement for augmented reality. *Computer Graphics Forum*, 15(3):C11–C22, September 1996.
4. Shenchang Eric Chen and Lance Williams. View interpolation for image synthesis. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 279–288, August 1993.
5. Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *Computer Graphics*, 22(4):75–84, August 1988. Proceedings SIGGRAPH '88 in Atlanta, USA.
6. P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In Holly Rushmeier, editor, *SIGGRAPH*, pages 11–20, New Orleans, August 1996.
7. Umesh R. Dhond and J.K. Aggarwal. Structure from stereo - a review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510, 1989.
8. George Drettakis and François Sillion. Interactive update of global illumination using a line-space hierarchy. In Turner Whitted, editor, *SIGGRAPH 97 Conference Proceedings (Los Angeles, CA)*, Annual Conference Series. ACM SIGGRAPH, August 1997.
9. Olivier Faugeras. On the evolution of simple curves of the real projective plane. Technical Report 1998, INRIA, 1993.
10. Olivier Faugeras. *Three-Dimensional Computer Vision: a Geometric Viewpoint*. MIT Press, 1993.
11. Olivier Faugeras and Stéphane Laveau. Representing three-dimensional data as a collection of images and fundamental matrices for image synthesis. In *Proceedings of the International Conference on Pattern Recognition*, pages 689–691, Jerusalem, Israel, October 1994. Computer Society Press.
12. Olivier Faugeras, Stéphane Laveau, Luc Robert, Gabriella Csurka, Cyril Zeller, Cyrille Gauclin, and Imed Zoghlami. 3-d reconstruction of urban scenes from image sequences. *CVGIP: Image Understanding*, 1997. To appear.
13. Olivier Faugeras, Tuan Luong, and Steven Maybank. Camera self-calibration: theory and experiments. In G. Sandini, editor, *Proc 2nd ECCV*, volume 588 of *Lecture Notes in Computer Science*, pages 321–334, Santa Margherita Ligure, Italy, May 1992. Springer-Verlag.
14. Alain Fournier, Atjeng S. Gunawan, and Chris Romanzin. Common illumination between real and computer generated scenes. In *Proceedings Graphics Interface '93*, pages 254–263. Morgan Kaufmann publishers, 1993.
15. Eric A. Haines. Shaft culling for efficient ray-traced radiosity. In Brunet and Jansen, editors, *Photorealistic Rendering in Comp. Graphics*, pages 122–138. Springer Verlag, 1993. Proc. 2nd EG Workshop on Rendering (Barcelona, 1991).
16. Pat Hanrahan, David Saltzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *Computer Graphics*, 25(4):197–206, August 1991. SIGGRAPH '91 Las Vegas.
17. R.I. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070, Boston, MA, June 1995. IEEE Computer Society Press.
18. M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *Proceedings of the 5th International Conference on Computer Vision*, pages 605–611, Boston, MA, June 1995. IEEE Computer Society Press.
19. P. Jancène, F. Neyret, X. Provot, J-P. Tarel, J-M. Vézien, C. Meilhac, and A. Vérroust. Res: computing the interactions between real and virtual objects in video sequences. In *2nd*

IEEE Workshop on networked Realities, pages 27–40, Boston, Ma (USA), October 1995. http://www-rocq.inria.fr/syntim/textes/nr95-eng.html.
20. C.P. Jerian and R. Jain. Structure from motion. a critical analysis of methods. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):572–587, 1991.
21. L. McMillan. Acquiring immersive visual environments with an uncalibrated camera. Technical Report TR95-006, University of North Carolina, 1995.
22. L Robert. Camera calibration without feature extraction. *Computer Vision, Graphics, and Image Processing*, 63(2):314–325, March 1995. also INRIA Technical Report 2204.
23. S.M. Seitz and C.R. Dyer. Physically-valid view synthesis by image interpolation. In *Proc. IEEE Workshop on Representation of Visual Scenes*, pages 18–25, Cambridge, Massachusetts, USA, June 1995.
24. François Sillion. A unified hierarchical algorithm for global illumination with scattering volumes and object clusters. *IEEE Trans. on Vis. and Comp. Graphics*, 1(3), September 1995.
25. Brian Smits, James Arvo, and Donald P. Greenberg. A clustering algorithm for radiosity in complex environments. In Andrew S. Glassner, editor, *SIGGRAPH 94 Conference Proceedings (Orlando, FL)*, Annual Conference Series, pages 435–442. ACM SIGGRAPH, July 1994.
26. Andrei State, Gentaro Hirota, David T. Chen, Bill Garrett, and Mark Livingston. Superior augmented reality registration by integrating landmark tracking and magnetic tracking. In Holly Rushmeier, editor, *SIGGRAPH 96 Conference Proceedings (New Orleans, LO)*, Annual Conference Series, pages 429–438. ACM SIGGRAPH, Addison Wesley, August 1996.
27. Richard Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, March 1996.
28. Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78(1-2):87–119, 1994.
29. I. Zoghlami, O. Faugeras, and R. Deriche. Using geometric corners to build a 2d mosaic from a set of images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition,* Puerto Rico, June 1997. IEEE.

7 Appendix: Epipolar geometry

In the general case of one or two cameras observing a non-planar scene from two different viewpoints, the three-dimensional geometry of the scene and of the cameras can be characterised by the *epipolar geometry* of the cameras, a purely projective geometric property which depends only on the configuration of the cameras. It tells us that given one point in one image, we can draw a line in the second image on which the corresponding point (i.e., the point representing the same physical point in space) necessarily lies. The epipolar geometry is captured by a 3×3 singular matrix called the *fundamental matrix* [13]: Two image points $\mathbf{m}_1, \mathbf{m}_2$ represent the same point in space if and only if

$$\mathbf{m}_2{}^{\mathrm{T}} \mathbf{F}_{12} \mathbf{m}_1 = 0 \tag{5}$$

The fundamental matrix is related to the intrinsic and extrinsic parameters of the two cameras:

$$\mathbf{A}_2^T \mathbf{F}_{12} \mathbf{A}_1 = [\mathbf{t}]_\times \mathbf{R} \tag{6}$$

where $\mathbf{R} = \mathbf{R}_1 \mathbf{R}_2^{\mathrm{T}}$, $\mathbf{t} = -\mathbf{R}_1 \mathbf{R}_2{}^{\mathrm{T}} \mathbf{t}_2 + \mathbf{t}_1$ represent the inter-camera motion and $[\mathbf{t}]_\times$ is the antisymmetric matrix such that $\forall \mathbf{x}, [\mathbf{t}]_\times \mathbf{x} = \mathbf{t} \times \mathbf{x}$.

The fundamental matrix can be computed from point correspondences in the images, without knowing anything about the intrinsic parameters of the cameras (focal length, aspect ratio, etc.). Robust programs which automatically perform this computation [28] are now publicly available[4].

[4] ftp://krakatoa.inria.fr/pub/robotvis/BINARIES

Fig. 5. The mosaic resulting from 3 original images. Textures are extracted using one image only.

Fig. 6. (a) The complete CAR rendering using the RVI texture polygons for display, including the CG object (b) The CG object has moved to the left: the update takes 2.5 seconds.